

ИССЛЕДОВАНИЕ АВТОЭНКОДЕРА ДЛЯ ВОССТАНОВЛЕНИЯ ИЗОБРАЖЕНИЯ И СИНТЕЗА НОВЫХ МЕДИЦИНСКИХ ДАННЫХ

В.В. Лаптев

Научный руководитель: О.М. Гергет
Томский политехнический университет, г. Томск
E-mail: lptwlad1@gmail.com

Введение

Машинное обучение — это применение искусственного интеллекта, предоставляющее возможность учиться и самосовершенствоваться на основе, получаемого, опыта без явного программирования.

В современной медицинской практике, в том числе и в области кардиологии, продолжает сохраняться устойчивый рост использования методов автоматической обработки графических данных. Наиболее популярными являются алгоритмы обработки анатомических структур на основе данных МРТ и КТ. Однако в ряде случаев использование представленных модальностей невозможно. Основным ограничением компьютерной томографии является отсутствие режима реального времени. Для решения данной проблемы необходима разработка и внедрение интеллектуального алгоритма трекинга и визуализации данных. Автоэнкодер позволит не только синтезировать данные для обучения модели трекинга, но восстанавливать изображения, убирая шум и прочие помехи. А также, благодаря сверточной архитектуре выделять и сохранять множество признаков (fiche, фичей).

Автоэнкодер

Для выполнения задач шумоподавления и снижения размерности данных в данной работе используются свёрточные автоэнкодеры. Автоэнкодеры представляют собой нейронные сети прямого распространения, которые восстанавливают входной сигнал на выходе (см. Рис. 1.). Внутри таких сетей имеется скрытый слой "Code", который отражает латентное представление, описывающее модель. Автоэнкодеры конструируются таким образом, чтобы не иметь возможность точно скопировать вход на выходе. Обычно их ограничивают в размерности латентного представления, которое меньше, чем размерность сигнала. Входной сигнал восстанавливается с ошибками из-за потерь при кодировании. Однако для того, чтобы их минимизировать, сеть вынуждена учиться отбирать наиболее важные признаки.

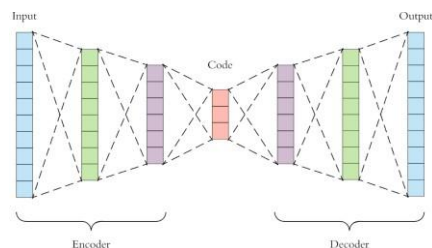


Рис. 1. Концептуальная схема автоэнкодера

Автоэнкодеры состоят из двух частей: энкодера g и декодера f . Энкодер переводит входной сигнал в его представление (код, латентное представление): $h = g(x)$, а декодер восстанавливает сигнал по его коду: $x = f(h)$. Тем самым автоэнкодер, изменяя f и g , стремится выучить тождественную функцию $x = f(g(x))$, минимизируя определённый функционал ошибки $L(x, f(g(x)))$. При этом семейства функций энкодера g и декодера f ограничены таким образом, чтобы автоэнкодер был вынужден отбирать наиболее важные свойства сигнала.

Исследование

В качестве одной из архитектур автоэнкодера использовался подход полносвязных слоёв (см. Рис. 2.). Общее количество обучаемых весов данной архитектуры составило 50 226 676.

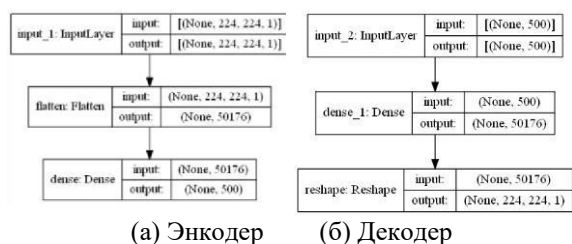


Рис. 2. Архитектура полносвязного автоэнкодера

При обучении автоэнкодера функционал ошибки L базировался на двух метриках, а именно на коэффициенте детерминации (R^2 сокр. от R-Squared) и перекрёстной энтропии (BCE сокр. от Binary Cross Entropy). В результате функционал ошибки L имел следующий вид:

$$L = R^2 + BCE \quad (1)$$

$$R^2 = 1 - \frac{SSR}{SST} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \quad (2)$$

$$BCE = -\frac{1}{N} \sum_{i=0}^N y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i) \quad (3)$$

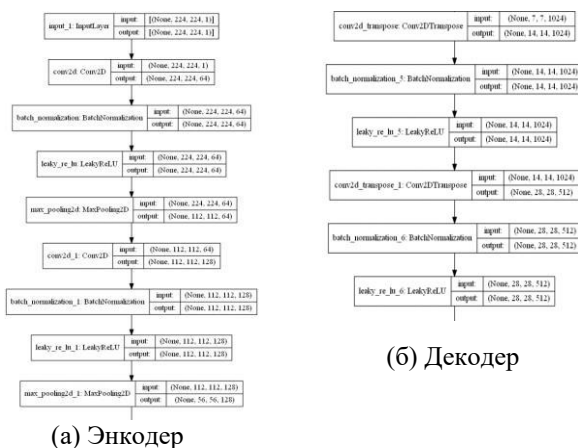
где y_i – значение наблюдения, \hat{y}_i – предсказанное значение наблюдения, \bar{y} – среднее значение всех наблюдений в выборке, N – количество наблюдений в выборке. Также дополнительно была оценена дивергенция Кульбака – Лейблера (KLD сокр. от Kullback–Leibler divergence), которая показывает информационное расхождение (относительную энтропию) двух вероятностных распределений и рассчитывается следующим образом:

$$KLD = \sum_{i=0}^N p(x_i) \cdot \log\left(\frac{p(x_i)}{q(x_i)}\right) \quad (4)$$

где $p(x_i)$ – исходное распределение, $q(x_i)$ – аппроксимирующее распределение.

При обучении модели, основанной на полносвязном автоэнкодере, функция потерь не сходилась. Последнее привело к слабой обобщающей способности модели.

Исходя из негативных результатов полносвязной архитектуры автоэнкодера, было принято решение реализовать собственную глубокую архитектуру сети (см. Рис. 3.), базирующейся на обучаемых сверточных и деконволюционных слоях. Стоит отметить, что глубокий сверточный автоэнкодер, включающий 5 уровней понижения размерности (на рисунке представлено 2 уровня, каждый последующий повторяет предыдущий), имел примерно в 2.5 раза меньше весов по сравнению с полносвязным автоэнкодером. Общее количество обучаемых весов данной архитектуры составило 21 991 937.



(а) Энкодер

(б) Декoder

Рис. 3. Архитектура глубокого сверточного автоэнкодера

Результат функции отражен на Рис. 4. Исходя из графиков видно, что модель сходится после 20 эпох, а переобучение (разница между значениями метрики на обучающей и валидационной выборках) нивелируется.

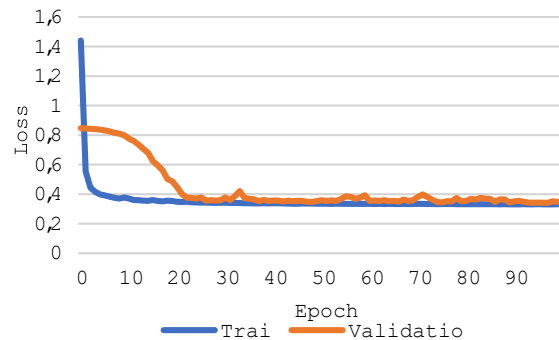


Рис. 4. Динамика функции потерь на этапах обучения и валидации модели

Заключение

Разработана собственная архитектура глубокого сверточного автоэнкодера, проведено сравнение с автоэнкодером, базирующемся на полносвязных слоях. Сравнение показало, что разработанный сверточный автоэнкодер выполняет реконструкцию данных эффективнее. Сверточный автоэнкодер реконструирует изображения примерно с 10% потерями. Тестирование данного автоэнкодера на искусственно зашумленных изображениях показало, что, имея лишь 40% информации от исходного изображения, он способен восстановить 75-80% информации. Также стоит отметить, что сверточная версия автоэнкодера практически в 2.5. раза менее требовательна по вычислительным ресурсам, чем полносвязный автоэнкодер.

Список использованных источников

1. Keras: The Python Deep Learning library. [Электронный ресурс] – Режим доступа: <https://keras.io>, свободный (дата обращения 13.09.19).
2. Autoencoders for Image Reconstruction in Python and Keras. [Электронный ресурс] – Режим доступа: <https://stackabuse.com/autoencoders-for-image-reconstruction-in-python-and-keras>, свободный (дата обращения 15.09.19).
3. TensorFlow is an end-to-end open source platform for machine learning. [Электронный ресурс] – Режим доступа: <https://www.tensorflow.org/tutorials>, свободный (дата обращения 13.09.19).
4. Как работает сверточная нейронная сеть: архитектура, примеры, особенности. [Электронный ресурс] – Режим доступа: <https://neurohive.io/ru/osnovy-datascience/glubokaya-svertochnaja-nejronnaja-set>, свободный (дата обращения 4.10.19)